MASK-BASED MODELING FOR NEURAL RADIANCE FIELDS (APPENDIX)

Ganlin Yang [1] **Guoqiang Wei** [2] **Zhizheng Zhang** [3]* **Yan Lu** [3] **Dong Liu** [1]*
[1] University of Science and Technology of China [2] ByteDance Research [3] Microsoft Research Asia
ygl666@mail.ustc.edu.cn weiguoqiang.9@bytedance.com
{zhizzhang,yanlu}@microsoft.com dongeliu@ustc.edu.cn

1 MORE EXPERIMENTAL RESULTS

1.1 RESULTS ON SYNTHETIC DATASETS

Category-agnostic ShapeNet-all and ShapeNet-unseen settings The overall numerical results have already been presented in the main paper. The detailed results with a breakdown by categories are provided in Table 1 and Table 2. We provide additional visual results in Figure 6, Figure 7 for **ShapeNet-all** setting and Figure 8, Figure 9 for **ShapeNet-unseen** setting, respectively. We randomly sample 4 object instances for each of the testing categories in ShapeNet dataset and show visual comparisons to PixelNeRF (Yu et al., 2021) and our baseline NeRFormer.

Category-specific ShapeNet-car and ShapeNet-chair settings The quantitative comparisons on PSNR, SSIM and LPIPS are available in the main paper. SRN (Sitzmann et al., 2019), FE-NVS (Guo et al., 2022) and CodeNeRF (Jang & Agapito, 2021) do not provide LPIPS result in their paper. We calculate LPIPS result for PixelNeRF (Yu et al., 2021) using author-provided checkpoints. More visualizations are shown in Figure 10 and Figure 11. We use view-64 and view-64, 104 as input view(s) for one-shot and two-shot cases. For each scenario we randomly sample 5 object instances, and show visual comparisons to PixelNeRF (Yu et al., 2021) and our baseline NeRFormer.

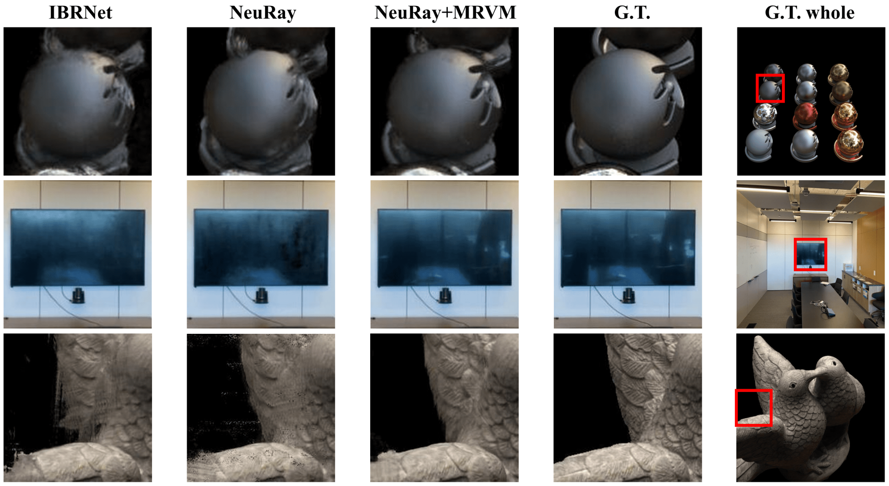

Figure 1: Visualizations for **cross-scene generalization** on NeRF Synthetic (first row), LLFF (middle row) and DTU (last row) datasets.

1.2 RESULTS ON REALISTIC DATASETS

For real-world **cross-scene generalization** and **per-scene finetuning** settings, as we illustrated in the main paper, we adopt NeuRay (Liu et al., 2022) as baseline and evaluate on three datasets: NeRF

*Corresponding authors: Z. Zhang and D. Liu

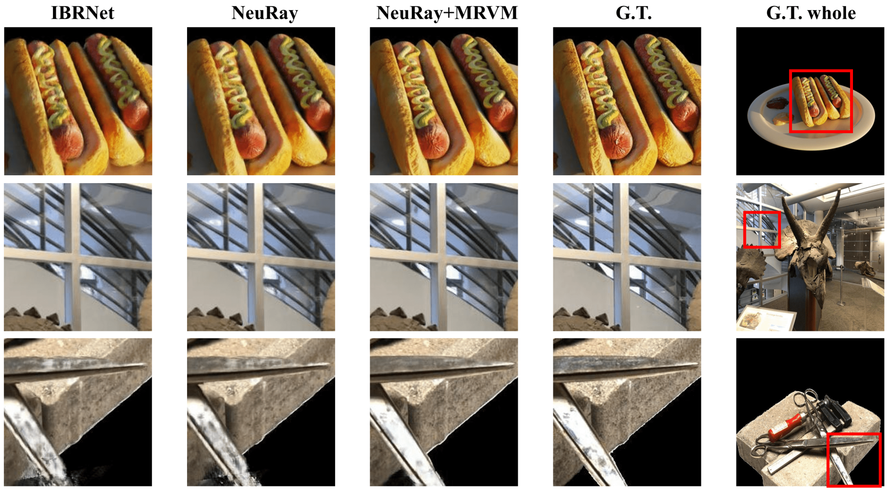

Figure 2: Visualizations for **per-scene finetuning** on NeRF Synthetic (first row), LLFF (middle row) and DTU (last row) datasets.

Synthetic (Niemeyer et al., 2020), DTU (Jensen et al., 2014) and LLFF (Mildenhall et al., 2019). The quantitative results are presented in Table 3 in the main paper, and more visualizations for **cross-scene generalization** setting and **per-scene finetuning** setting are shown in Figure 1 and Figure 2 respectively.

Table 1: Detailed results of **category-agnostic ShapeNet-all** setting, with a breakdown by categories. This table is an expansion of Table 1 in the main paper.

Metric	Method	plane	bench	cbnt.	car	chair	disp.	lamp	spkr.	rifle	sofa	table	phone	boat	avg.
PSNR↑	SRN	26.62	22.20	23.42	24.40	21.85	19.07	22.17	21.04	24.95	23.65	22.45	20.87	25.86	23.28
	PixelNeRF	29.76	26.35	27.72	27.58	23.84	24.22	28.58	24.44	30.60	26.94	25.59	27.13	29.18	26.80
	FE-NVS	30.15	27.01	28.77	27.74	24.13	24.13	28.19	24.85	30.23	27.32	26.18	27.25	28.91	27.08
	SRT	31.47	28.45	30.40	28.21	24.69	24.58	28.56	25.61	30.09	28.11	27.42	28.28	29.18	27.87
	VisionNeRF	32.34	29.15	31.01	29.51	25.41	25.77	29.41	26.09	31.83	28.89	27.96	29.21	30.31	28.76
	NeRFormer	30.50	27.19	28.88	28.12	24.49	25.21	29.34	25.22	31.13	27.65	26.67	27.93	30.12	27.58
	NeRFormer+MRVM	32.10	28.91	30.94	29.16	26.20	27.27	31.54	27.24	32.18	29.25	28.82	29.70	31.13	**29.25**
SSIM↑	SRN	0.901	0.837	0.831	0.897	0.814	0.744	0.801	0.779	0.913	0.851	0.828	0.811	0.898	0.849
	PixelNeRF	0.947	0.911	0.910	0.942	0.858	0.867	0.913	0.855	0.968	0.908	0.898	0.922	0.939	0.910
	FE-NVS	0.957	0.930	0.925	0.948	0.877	0.871	0.916	0.869	0.970	0.920	0.914	0.926	0.941	0.920
	SRT	0.954	0.925	0.920	0.937	0.861	0.855	0.904	0.854	0.962	0.911	0.909	0.918	0.930	0.912
	VisionNeRF	0.965	0.944	0.937	0.958	0.892	0.891	0.925	0.877	0.974	0.930	0.929	0.936	0.950	0.933
	NeRFormer	0.953	0.921	0.922	0.947	0.870	0.879	0.924	0.869	0.971	0.916	0.913	0.928	0.946	0.920
	NeRFormer+MRVM	0.966	0.945	0.941	0.958	0.906	0.912	0.948	0.900	0.978	0.937	0.942	0.944	0.959	**0.942**
LPIPS↓	SRN	0.111	0.150	0.147	0.115	0.152	0.197	0.210	0.178	0.111	0.129	0.135	0.165	0.134	0.139
	PixelNeRF	0.084	0.116	0.105	0.095	0.146	0.129	0.114	0.141	0.066	0.116	0.098	0.097	0.111	0.108
	FE-NVS	0.061	0.080	0.076	0.085	0.103	0.105	0.091	0.116	0.048	0.081	0.071	0.080	0.094	0.082
	SRT	0.050	0.068	0.058	0.062	0.085	0.087	0.082	0.096	0.045	0.066	0.055	0.059	0.079	0.066
	VisionNeRF	0.042	0.067	0.065	0.059	0.084	0.086	0.073	0.103	0.046	0.068	0.055	0.068	0.072	0.065
	NeRFormer	0.063	0.096	0.088	0.081	0.128	0.116	0.093	0.126	0.055	0.099	0.079	0.083	0.090	0.091
	NeRFormer+MRVM	0.045	0.067	0.064	0.059	0.087	0.083	0.065	0.098	0.042	0.070	0.051	0.063	0.070	**0.060**

1.3 Results on other baselines

We also provide the additional experimental results of adding our proposed masked ray and view modeling (MRVM) on another advanced generalizable NeRF baseline GNT (Wang et al., 2022), on NeRF Synthetic (Niemeyer et al., 2020) and LLFF (Mildenhall et al., 2019) datasets respectively, and compare with another state-of-the-art method GNT-MOVE (Cong et al., 2023). The default setting for novel-view synthesis is put in Table 3 and the few-shot setting is located in Table 4. We conclude that the proposed masked ray and view modeling consistently benefits under all the cases.

Table 2: Detailed results of **category-agnostic ShapeNet-unseen** setting, with a breakdown by categories. This table is an expansion of Table 1 in the main paper.

Metric	Method	bench	cbnt.	disp.	lamp	spkr.	rifle	sofa	table	phone	boat	avg.
PSNR↑	SRN	18.71	17.04	15.06	19.26	17.06	23.12	18.76	17.35	15.66	24.97	18.71
	PixelNeRF	23.79	22.85	18.09	22.76	21.22	23.68	24.62	21.65	21.05	26.55	22.71
	FE-NVS	23.10	22.27	17.01	22.15	20.76	23.22	24.20	20.54	19.59	25.77	21.90
	NeRFormer	23.64	22.21	17.77	23.20	20.60	24.11	24.58	21.05	21.24	27.32	22.54
	NeRFormer+MRVM	25.46	23.28	18.72	24.79	21.93	25.19	26.63	22.61	21.78	28.54	**24.08**
SSIM↑	SRN	0.702	0.626	0.577	0.685	0.633	0.875	0.702	0.617	0.635	0.875	0.684
	PixelNeRF	0.863	0.814	0.687	0.818	0.778	0.899	0.866	0.798	0.801	0.896	0.825
	FE-NVS	0.865	0.819	0.686	0.822	0.785	0.902	0.872	0.792	0.796	0.898	0.825
	NeRFormer	0.863	0.808	0.689	0.837	0.774	0.908	0.875	0.786	0.817	0.914	0.826
	NeRFormer+MRVM	0.892	0.815	0.693	0.857	0.786	0.921	0.899	0.822	0.827	0.927	**0.849**
LPIPS↓	SRN	0.282	0.314	0.333	0.321	0.289	0.175	0.248	0.315	0.324	0.163	0.280
	PixelNeRF	0.164	0.186	0.271	0.208	0.203	0.141	0.157	0.188	0.207	0.148	0.182
	FE-NVS	0.135	0.156	0.237	0.175	0.173	0.117	0.123	0.152	0.176	0.128	0.150
	NeRFormer	0.141	0.175	0.243	0.181	0.185	0.109	0.127	0.177	0.182	0.101	0.159
	NeRFormer+MRVM	0.096	0.135	0.220	0.135	0.148	0.082	0.088	0.115	0.146	0.089	**0.117**

Table 3: Experimental results of adding our proposed masked ray and view modeling on the baseline of GNT (Wang et al., 2022) and compare with GNT-MOVE (Cong et al., 2023) on NeRF Synthetic and LLFF datasets.

Method	Synthetic Object NeRF			Real Forward-facing LLFF		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
GNT	27.29	0.937	0.056	25.86	0.867	0.116
GNT-MOVE	27.47	0.940	0.056	26.02	0.869	**0.108**
GNT+MRVM	**27.78**	**0.942**	**0.052**	**26.25**	**0.873**	0.110

2 More Implementation Details

We first provide general configurations that are applicable across all settings, followed by configurations specific to each unique setting.

General configurations For mask-based pretraining, we incorporate $\mathcal{L}_{mrvm}$ as an auxiliary loss. It is optimized together with NeRF's rendering loss not from the beginning, but starting from 10% of the total training iterations until finishing. We also use a warm-up schedule for about 10k iterations which linearly increases the coefficient λ from 0 to the final value 0.1. Both of these technical strategies contribute to stabilize the pretraining process. At inference time, we use the VGG network for calculating LPIPS (Zhang et al., 2018) after normalizing pixel values to [-1,1]. We perform ray casting, sampling and volume rendering all in the world coordinate. All the models are implemented using Pytorch (Paszke et al., 2019) framework.

2.1 Implementation Details for synthetic datasets

Considering the images of synthetic datasets have a blank background, we adopt two techniques following previous works (Yu et al., 2021; Lin et al., 2022) for better performance. 1) We use bounding box sampling strategy as Yu et al. (2021) during pretraining, where rays are only sampled within the bounding box of the foreground object. In this way, it avoids the model to learn *too much empty* information at initial training stage. 2) We assign a white background color for those pixels sampled from the background to match the rendering ground truths in ShapeNet dataset.

Settings For category-agnostic **ShapeNet-all** setting, we use a batch size of 16, and sample 256 rays per object. We pretrain the model for 400k iterations on 4 GPUs, with a tight bounding box for the first 300k iterations, then we finetune the model without bounding box for 800k iterations. The two-stage training takes about 10 days on GTX-1080Ti.

Table 4: The few-shot experimental results of adding our proposed masked ray and view modeling on the baseline of GNT (Wang et al., 2022) and compare with GNT-MOVE (Cong et al., 2023) on NeRF Synthetic and LLFF datasets.

Method	Synthetic Object NeRF						Real Forward-facing LLFF					
	6-shot			12-shot			3-shot			6-shot		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
GNT	22.39	0.856	0.139	25.25	0.901	0.088	19.58	0.653	0.279	22.36	0.766	0.189
GNT-MOVE	22.53	**0.871**	**0.116**	25.85	**0.915**	**0.074**	19.71	0.666	0.270	22.53	0.774	0.184
GNT+MRVM	**23.52**	0.869	0.120	**26.10**	0.911	0.079	**20.88**	**0.672**	**0.257**	**23.54**	**0.777**	**0.175**

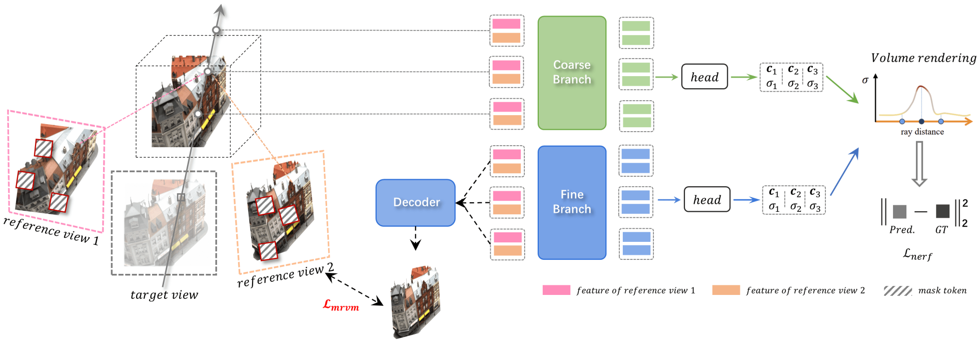

Figure 3: Illustration for mask-based pretraining variant 1 — **RGB mask**. We mask blocks of pixels and try to recover them at pretraining.

For category-agnostic **ShapeNet-unseen** setting, we also use a batch size of 16, and sample 256 rays per object. We pretrain for 300k iterations with bounding box on 4 GPUs, and finetune the model for 600k iterations without bounding box, which takes about 8 days on GTX-1080Ti.

For category-specific **ShapeNet-car** and **ShapeNet-chair** settings, we use a batch size of 8, and sample 512 rays per object. We pretrain for 400k iterations on 4 GPUs. For the first 300k iterations, we use 2 input views for the network to encode with a tight bounding box. For the rest of 100k iterations, the bounding box is removed and we randomly choose 1 or 2 view(s) as the input to make the model compatible with both one-shot and two-shot scenarios. We finetune the model for 1 or 2 view(s) respectively on 8 GPUs for 400k iterations. The two-stage training takes about 7 days on GTX-1080Ti.

2.2 Implementation Details for realistic datasets

Following the training protocol in NeuRay (Liu et al., 2022), we first perform cross-scene pretraining across five distinct datasets (Downs et al., 2022; Mildenhall et al., 2019; Flynn et al., 2019; Zhou et al., 2018; Jensen et al., 2014) for 400k iterations. Afterwards, for **cross-scene generalization** setting, we finetune the model on the same five training sets for additional 200k iterations. For **per-scene finetuning** setting, the model is finetuned on each scene respectively in the three testing datasets (Niemeyer et al., 2020; Jensen et al., 2014; Mildenhall et al., 2019) for additional 100k iterations, except for the few-shot scenarios in Table 5 of the main paper where we find only 10k iterations is sufficient for finetuning. When training the generalizable model across multiple datasets, we randomly sample 1 scene from the training sets per iteration. We sample 512 rays for each scene during training. All the training processes are conducted on one V100 GPU, which takes about 5 days for total pretraining and finetuning.

2.3 Variants of Mask-Based Pretraining Objectives

As stated in the main paper, we conduct an elaborated ablation study on different mask-based pretraining strategies, which are illustrated in Figure 3, Figure 4 and Figure 5.

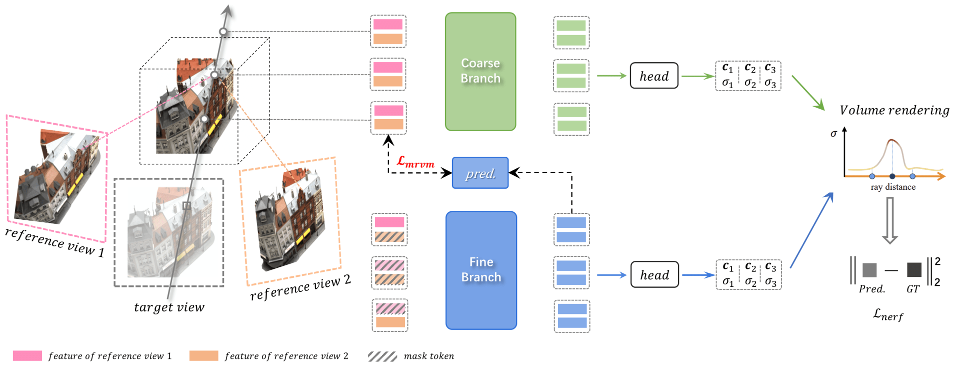

Figure 4: Illustration for mask-based pretraining variant 2 — **Feat mask^1:**. We use the intermediate representation output (boxes in blue) by Fine-Branch to reconstruct the masked feature tokens.

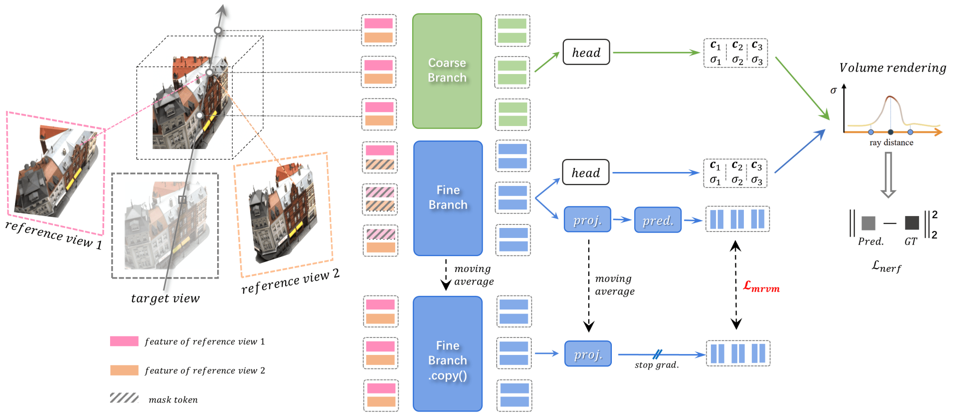

Figure 5: Illustration for mask-based pretraining variant 3 — **Feat mask^2:**. We make a copy of Fine-Branch as the target branch, in place of Coarse-Branch in the main paper.

- **RGB mask:** As shown in Figure 3, we mask blocks of pixels on input images from reference views. After extracting pyramid features with a 2D CNN, we additionally introduce an UNet-like decoder to recover the masked image pixels based on these features. $\mathcal{L}_{mrvm}$ is the $\mathcal{L}_2$ distance between reconstructed pixels and the ground truth, the constraint is only added to masked regions. We set mask ratio to 50% and patch size to 4 at pretraining.
- **Feat mask^1:** As illustrated in Figure 4, we perform masking operation on sampled points same as MRVM. Differently, after obtaining intermediate representation $\mathbf{z}_i^j$ from the fine branch, we use it to recover the masked latent feature $\mathbf{h}_i^j$ by a shallow 2-layer MLP. $\mathcal{L}_{mrvm}$ is the $\mathcal{L}_2$ distance between the reconstructed latent feature vector and the unmasked ground truth. We normalize the vector to unit-length before calculating the distance.
- **Feat mask^2:** The pipeline for this variant is presented in Figure 5. Different from the architecture in the main paper, we don't utilize coarse branch as the target. On the contrary, we make a copy of the fine branch as the target network. With the gradient stopped manually, this branch is updated by moving average of the parameters from the online fine branch. We experimentally find that this option may cause instability at mask-based pretraining stage, making it inappropriate as our final proposal.

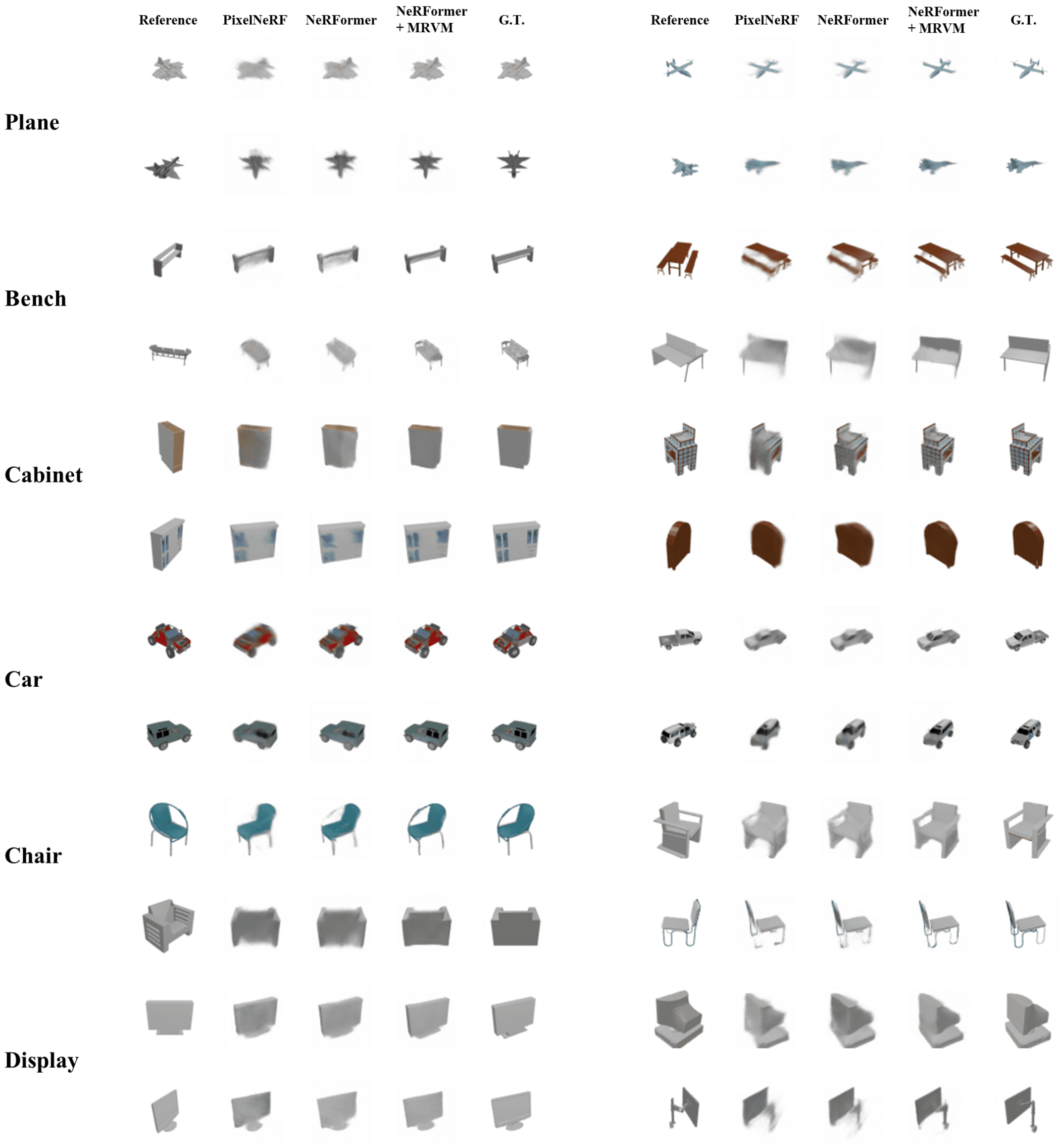

Figure 6: More visualizations for **Category-agnostic ShapeNet-all** setting, Part 1.

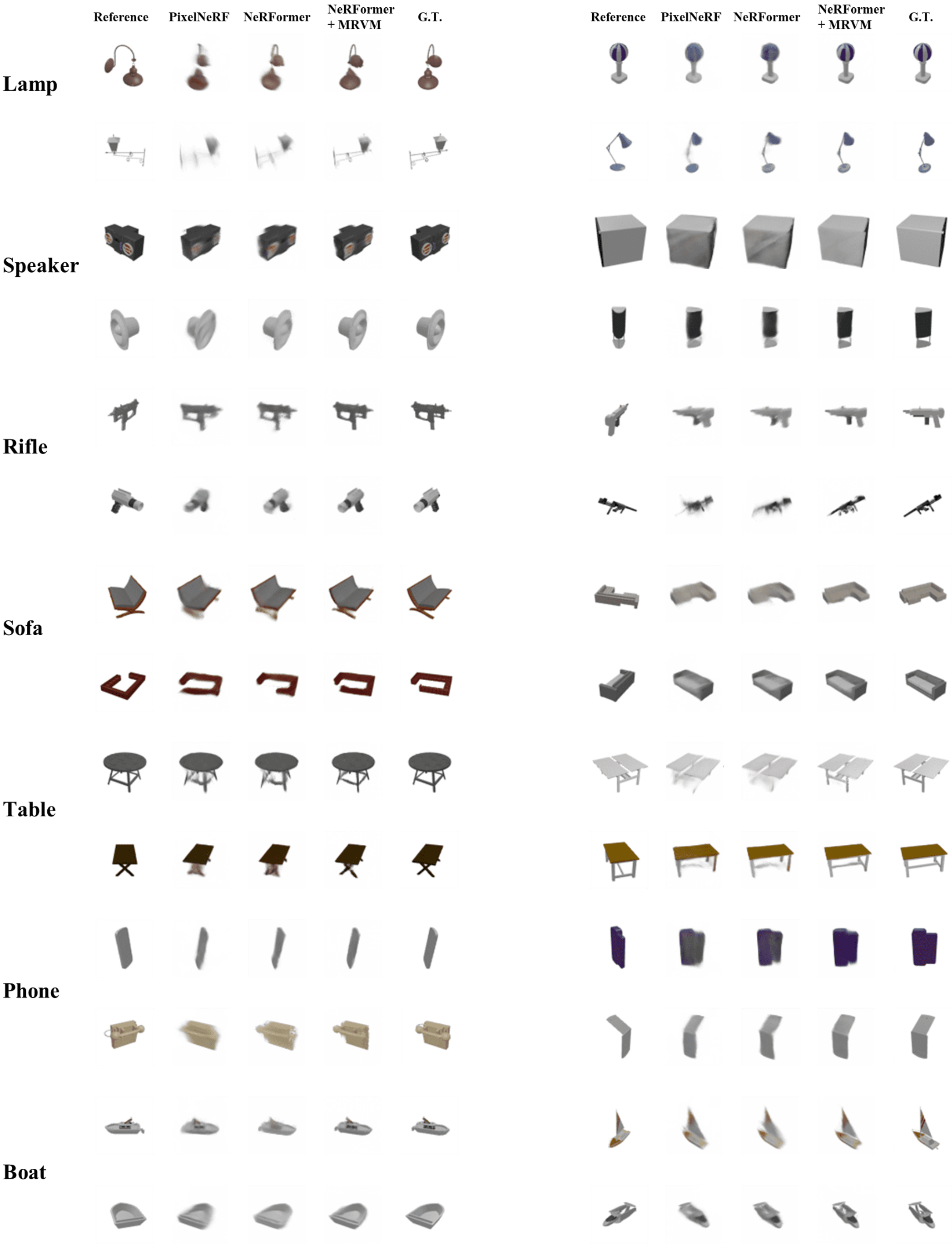

Figure 7: More visualizations for **Category-agnostic ShapeNet-all** setting, Part 2.

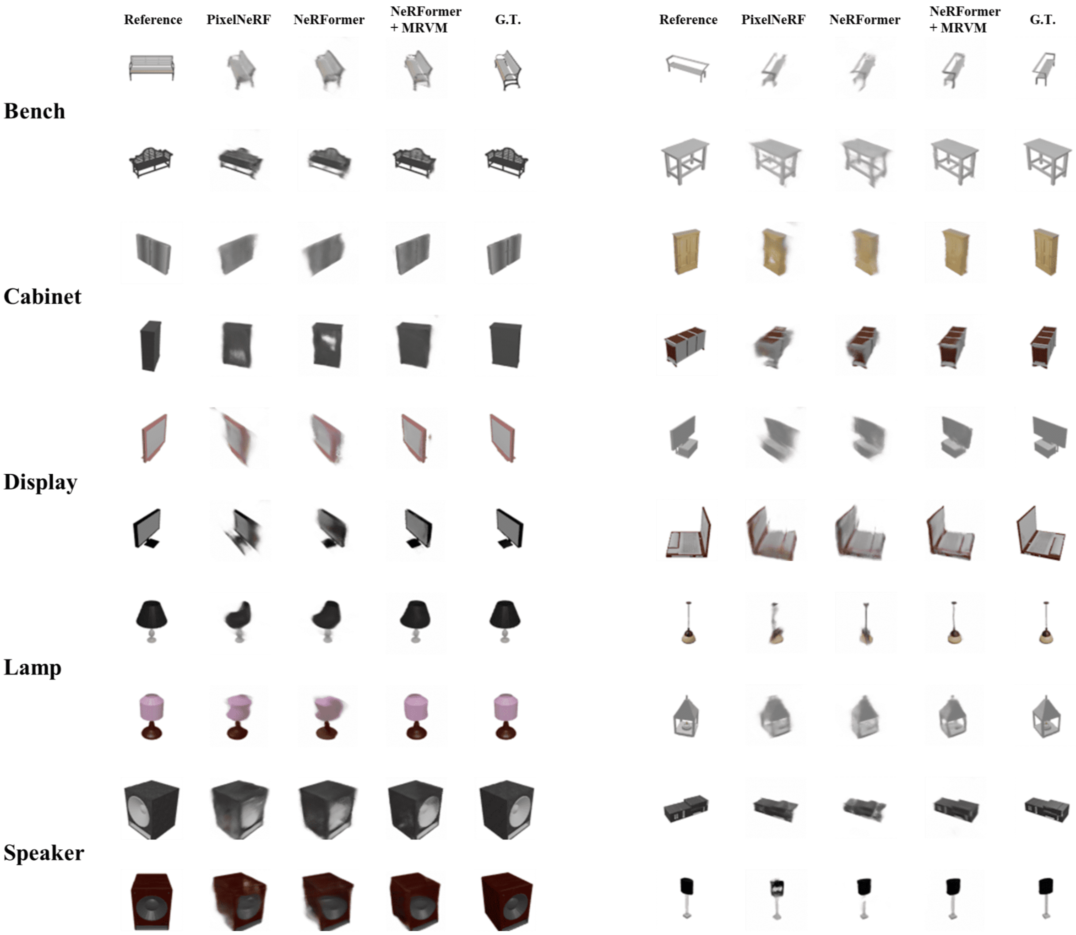

Figure 8: More visualizations for **Category-agnostic ShapeNet-unseen** setting, Part 1.

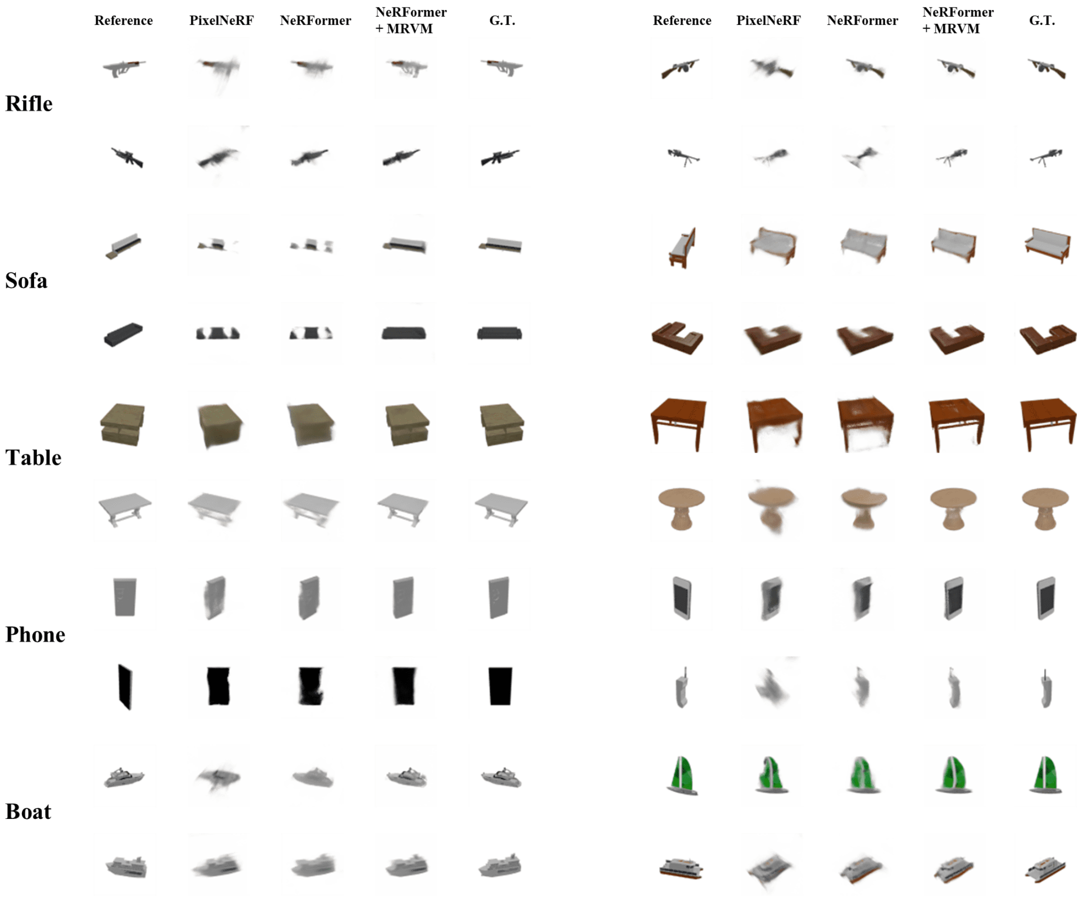

Figure 9: More visualizations for **Category-agnostic ShapeNet-unseen** setting, Part 2.

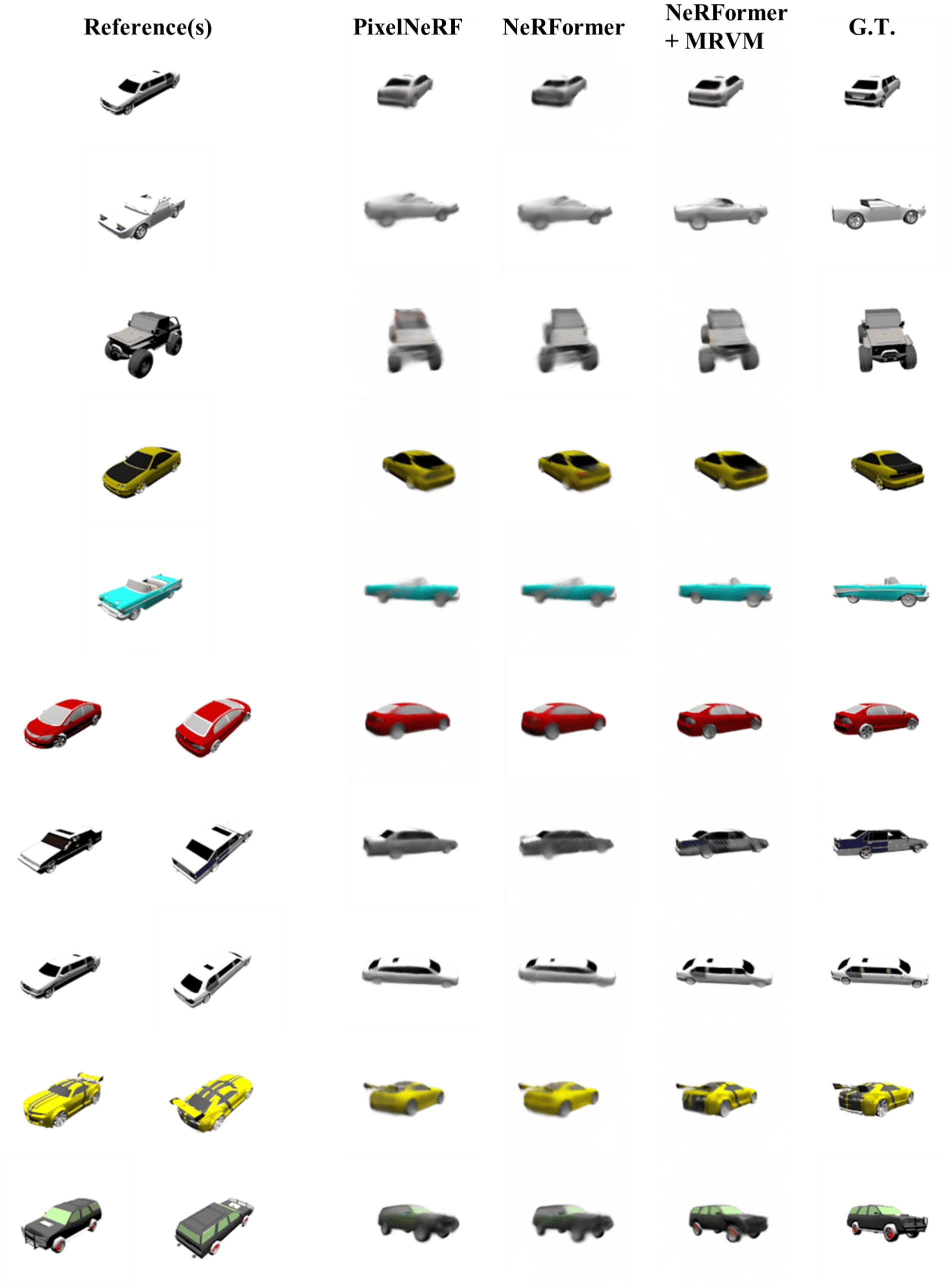

Figure 10: More visualizations for **Category-specific ShapeNet-car** setting.

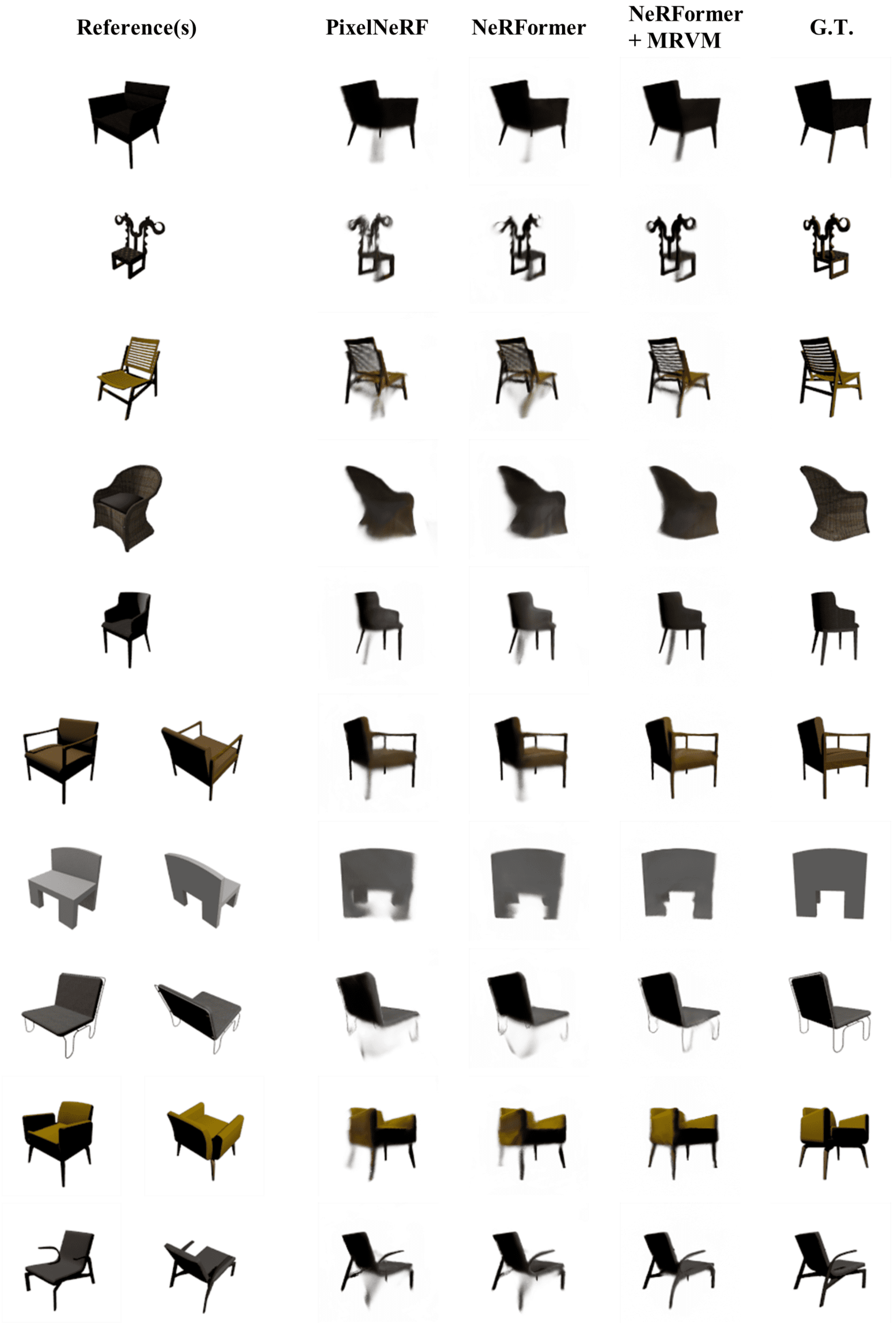

Figure 11: More visualizations for **Category-specific ShapeNet-chair** setting.

References

Wenyan Cong, Hanxue Liang, Peihao Wang, Zhiwen Fan, Tianlong Chen, Mukund Varma, Yi Wang, and Zhangyang Wang. Enhancing nerf akin to enhancing llms: Generalizable nerf transformer with mixture-of-view-experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3193–3204, 2023.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560. IEEE, 2022.

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2367–2376, 2019.

Pengsheng Guo, Miguel Angel Bautista, Alex Colburn, Liang Yang, Daniel Ulbricht, Joshua M Susskind, and Qi Shan. Fast and explicit neural view synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3791–3800, 2022.

Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12949–12958, 2021.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 406–413, 2014.

Kai-En Lin, Lin Yen-Chen, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. *arXiv preprint arXiv:2207.05736*, 2022.

Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7824–7833, 2022.

Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, Zhangyang Wang, et al. Is attention all that nerf needs? *arXiv preprint arXiv:2207.13298*, 2022.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.